Aide AppleScript

Essentiels

À propos d’AppleScript

À mesure que vous utilisez votre Mac, vous découvrirez que certaines tâches sont répétitives. AppleScript vous permet d’écrire des scripts pour contrôler une ou plusieurs applications et automatiser des tâches.

AppleScript est un langage de script puissant et polyvalent intégré à Mac OS X. Il vous permet de créer des raccourcis, d’automatiser des tâches répétitives ou même de créer des applications personnalisées qui vous feront gagner beaucoup de temps. AppleScript vous permet de contrôler les applications pilotables par script.

Un script AppleScript est composé de commandes proches de l’anglais, exécutant certaines actions sur des objets. Parmi les actions disponibles, on trouve « open » (ouvrir), « close » (fermer) et « get the name of » (obtenir le nom de). Les objets peuvent être une application, une fenêtre ou un réglage de préférences.

Un script permet de contrôler une ou plusieurs applications. Pour qu’un script puisse être utilisé sur une application, cette dernière doit définir les termes que vous pouvez utiliser dans vos scripts. Mac OS X contient plusieurs applications pilotables.

Les scripts permettent aussi de contrôler des applications sur un réseau. Vous pouvez ainsi automatiser des tâches sur plusieurs ordinateurs.

En fait, vous utilisez AppleScript sans le savoir. De nombreuses fonctions dans le système et dans diverses applications sont en effet des scripts AppleScript appelés par une application. Par exemple, dans le système d’aide, c’est un script AppleScript qui s’exécute lorsque vous cliquez sur un lien ouvrant une application ou un fichier.

RUBRIQUES CONNEXES :

Fonctionnement d’AppleScript
À propos de l’Éditeur AppleScript
À propos du langage AppleScript
Ressources AppleScript sur Internet
Environnement de développement AppleScript Studio

Essentiels

À propos de l’Éditeur AppleScript

L’Éditeur AppleScript est une application d’écriture de scripts permettant de lire, écrire et enregistrer des scripts AppleScript.

Voici quelques-unes des fonctionnalités de l’Éditeur AppleScript :

- Offre une interface utilisateur unifiée plus facile à utiliser.
- Permet de rechercher et remplacer des éléments dans le texte des scripts.
- Ajuste automatiquement les instructions longues dans les scripts.

- Permet d’effectuer plusieurs annulations.
- Offre des scripts de menu contextuel pour accélérer l’écriture des scripts.
- Prend en charge les services d’application.
- Permet d’accéder à des scripts d’application à l’aide de la palette de bibliothèque
- Fournit un Assistant Script pour vous aider à rédiger les instructions.
- Fournit un historique de résultat et un historique des événements afin que vous puissiez examiner le déroulement d’un script.

RUBRIQUES CONNEXES :

À propos du langage AppleScript
Environnement de développement AppleScript Studio

Essentiels

Créer des scripts AppleScript

Vous n’avez pas besoin de savoir programmer pour tirer parti d’AppleScript. En réalité, il ne vous est même pas nécessaire d’écrire un mot de script. Il existe des centaines de scripts AppleScript très pratiques, inclus dans OS X, que vous pouvez commencer par utiliser. Des milliers d’autres sont disponibles sur le web.

Si vous souhaitez aller un peu plus loin, vous pouvez utiliser des scripts existants et les adapter à vos besoins en y apportant des modifications mineures.

Bien évidemment, vous tirerez le meilleur parti d’AppleScript en créant vos propres scripts. Si vous apprenez à les écrire vous-même, vous serez en mesure de créer des outils puissants et des automatisations qui vous feront gagner du temps et amélioreront ainsi votre productivité.

Vous pouvez modifier ou créer vos propres scripts avec l’Éditeur AppleScript qui offre un environnement simple pour l’écriture de scripts. Tous les exemples de l’Aide AppleScript ont été codés par le biais de l’Éditeur AppleScript.

- Ouvrez Éditeur AppleScript.

  Tapez le texte suivant dans la fenêtre de l’Éditeur AppleScript.

  ```
  say "Welcome to AppleScript!"
  ```

  Cliquez sur Exécuter.

RUBRIQUES CONNEXES :

À propos de l’Éditeur AppleScript
Environnement de développement AppleScript Studio

Essentiels

Fonctionnement d’AppleScript

AppleScript fonctionne en envoyant des messages appelés « Apple events ». Lorsque vous exécutez un script, AppleScript

envoie des Apple events au système ou aux applications qui exécutent alors les commandes.

La plupart du temps, les applications ou le système renvoient un résultat à AppleScript. Ainsi, si un script envoie une requête à une application pour obtenir le nom d'un objet, l'application lui renvoie le nom de cet objet. Si la commande ne peut pas être exécutée, l'application ou AppleScript renvoie souvent un message d'erreur. Vous pouvez visualiser les résultats dans l'Éditeur AppleScript.

En règle générale, il n'est pas nécessaire de connaître les mécanismes régissant les Apple Events. AppleScript s'occupe de tout cela en arrière-plan. Vous n'avez donc besoin que d'écrire et d'exécuter vos scripts.

RUBRIQUES CONNEXES :

À propos d'AppleScript
À propos du langage AppleScript

Essentiels

À propos des applets

Un script enregistré en tant qu'application s'appelle une application ou un applet AppleScript. Les applets s'exécutent automatiquement lors de leur ouverture et exécutent les commandes contenues dans le script. Les applets peuvent aussi bien être de petites fonctions composées de seulement quelques commandes ou des applications complexes contenant plusieurs milliers de lignes.

Les applets peuvent être enregistrés selon deux modes : fichier exécutable seulement ou modifiable. Un applet modifiable conserve le texte du script pour qu'il soit possible de le modifier et de recompiler à tout moment. Pour éditer ou changer un applet, faites-le glisser sur l'icône Éditeur AppleScript. Lorsque vous enregistrez un script en fichier exécutable seulement, la portion de texte modifiable est supprimée et le fichier ne peut plus être modifié ou recompilé.

Pour modifier un applet ayant été enregistré afin d'être modifiable, ouvrez-le dans l'Éditeur AppleScript ou dans un autre éditeur AppleScript. Les applets nécessitent un gestionnaire « on run » qui lance l'exécution du script lorsque l'applet est ouvert.

- L'exemple suivant illustre un gestionnaire « on run » :

```
on run
        --première commande
        --deuxième commande
end run
```

RUBRIQUES CONNEXES :

À propos des droplets
Types de fichiers AppleScript

Essentiels

À propos des scripts compilés

Un script compilé est un fichier AppleScript sous une forme modifiable que vous pouvez lire et modifier dans l'Éditeur AppleScript ou dans un autre éditeur AppleScript.

Lorsque vous ouvrez un script compilé, celui-ci apparaît dans l'Éditeur AppleScript mais n'exécute aucune des commandes automatiquement. Ceci vous permet d'effectuer des modifications sur le script, sans l'exécuter. Vous pouvez alors compiler le script afin qu'AppleScript lise et convertisse tout le texte correspondant à du code AppleScript dans le fichier. Le script compilé peut être exécuté. Un script compilé possède l'extension de fichier .scpt.

Lorsque vous sauvegardez le fichier en tant que script compilé, le texte demeure dans le fichier avec ainsi que le code d'exécution. Le script compilé peut donc être modifié à tout moment.

RUBRIQUES CONNEXES :

À propos des applets
À propos des droplets
Types de fichiers AppleScript

Essentiels

À propos des droplets

Un droplet est un script AppleScript qui s'exécute lorsque vous faites glisser des fichiers ou des dossiers sur son icône. Le script exécute des actions sur ces fichiers ou ces dossiers.

Tout comme pour les applets, un droplet peut être enregistré selon deux modes : modifiable ou fichier exécutable seulement. Un droplet modifiable contient encore le texte du script, pour qu'il soit possible de le modifier et de recompiler à tout moment. Pour éditer ou changer un droplet, faites-le glisser sur l'icône Éditeur de scripts. Lorsque vous enregistrez un script en fichier exécutable seulement, la portion de texte modifiable est supprimée et le fichier ne peut plus être modifié ou recompilé.

Les droplets nécessitent un gestionnaire « on open » qui lance l'exécution du script lorsque le droplet est ouvert. Ce gestionnaire possède un paramètre qui contient une liste des éléments déposés sur le droplet. Une partie du script de droplet examine la liste, tandis que l'autre agit sur les éléments. Sauf si le droplet est uniquement conçu pour agir sur un seul élément, il doit contenir une boucle de répétition qui traite chaque élément de la liste, exécutant des actions sur chacun des éléments, tour à tour.

- Voici un exemple de gestionnaire de droplet avec l'instruction « on open » et la liste de paramètres :

```
on open (theItemsDropped)
        --première commande
        --deuxième commande
end open
```

RUBRIQUES CONNEXES :

À propos des applets
À propos des scripts compilés
Types de fichiers AppleScript

Essentiels

Utiliser des dictionnaires AppleScript

Pour savoir quels sont les commandes et les objets que vous pouvez utiliser pour contrôler une application, vous pouvez

consulter le dictionnaire de cette dernière. Un dictionnaire contient un ensemble de termes compris par une application.

Vous pouvez afficher le dictionnaire d'une application dans l'Éditeur AppleScript en sélectionnant l'application dans la fenêtre Bibliothèque ou en faisant glisser l'icône de l'application sur l'icône de l'Éditeur AppleScript.

Dans un dictionnaire, les termes sont organisés en groupes de termes liés, appelés « suites ». Les commandes et les objets sont en gras.

---

1. Dans l'Éditeur AppleScript, choisissez Fenêtre > Bibliothèque.
2. Double-cliquez sur le nom de l'application dans la fenêtre Bibliothèque.
3. Sélectionnez une suite.

   Les termes de la suite ainsi que leur description s'affichent.

Le langage AppleScript se compose de différentes catégories :

- Les suites : groupes de termes connexes. Dans le dictionnaire de l'Éditeur AppleScript, les suites sont affichées dans la première colonne et sont désignées par la lettre « S » orange.
- Les commandes : demandent l'exécution d'une action ou un résultat. Les commandes, qui sont également nommées « événements », sont affichées dans la deuxième colonne et sont désignées par la lettre « C » bleue (également nommée événement.)
- Les classes : objets pilotables. Les descriptions de classes dressent une liste des propriétés et des éléments qui composent une classe. Les classes sont affichées dans la deuxième colonne et sont désignées par la lettre « C » mauve.
- Les propriétés : objets contenant des valeurs. Les propriétés sont affichées dans la troisième colonne et sont désignées par la lettre « P » mauve.
- Les éléments : objets contenus dans d'autres objets. Par exemple, un paragraphe peut contenir de nombreux mots. Les éléments peuvent apparaître dans la troisième colonne et sont désignés par la lettre « E » orange.

RUBRIQUES CONNEXES :

Définitions de classes
Définitions de commandes
Utiliser AppleScript pour piloter des applications

OUVRIR :

Guide du langage AppleScript : Référence sur les commandes
Guide du langage AppleScript : Référence sur les classes

Essentiels

Résoudre les problèmes

Si vous rencontrez des problèmes lors de l'utilisation d'AppleScript, parcourez les rubriques suivantes pour obtenir des informations vous permettant de résoudre ces problèmes.

Si un message d'erreur lorsqu'un utilisateur clique sur Annuler dans une fenêtre Choisir dans la liste

Si votre script d'action de dossier s'est déclenché inopinément

Si votre script ne reconnaît pas un disque

Définitions de classes

Une définition de classe décrit les propriétés et les éléments d'un objet pilotable. Une classe peut être comparée à un nom dans une phrase et la définition de la classe est analogue à une définition de dictionnaire pour ce nom.

Une définition de classe dans un dictionnaire AppleScript commence généralement par le terme en gras suivi d'une brève description du terme sur la même ligne.

Elle est suivie d'une liste d'éléments ou d'objets appartenant à l'objet pilotable ou comprenant ce dernier. Une liste d'éléments n'est pas toujours incluse dans toutes les définitions de classe, car certains objets pilotables contiennent d'autres objets pilotables alors que ce n'est pas le cas pour d'autres.

La définition de classe se termine par une liste de propriétés, ou qualités, de l'objet pilotable et de leurs valeurs.

Une entrée de propriété commence par le nom de la propriété suivi de :

- Une description du format de données sous lequel la valeur de la propriété est stockée, par exemple : « integer » (nombre entier), « point » (une liste de deux nombres), « string » (données texte), « Boolean » (vrai ou faux).
- Un indicateur de type optionnel, [r/o], qui signale que la valeur de la propriété est « read-only » (en lecture seule), c'est-à-dire qu'elle peut être lue, mais pas modifiée.
- Une brève description de la propriété.

Si un objet a une forme plurielle, il vous permet de faire référence à plus d'un objet.

Si un objet fait possède une liste d'éléments, cela signifie qu'il contient d'autres objets. Le nom de l'élément, que vous pouvez utiliser dans votre script, apparaît en gras. Le texte standard décrit la méthode permettant de faire référence à l'élément.

Les objets peuvent avoir des propriétés. Le nom des propriétés apparaît en gras. Le texte standard décrit le type de valeur que contient la propriété, telle qu'une chaîne de caractères. Les propriétés peuvent être une valeur constante ou une liste de constantes parmi plusieurs possibles. Un script peut obtenir la valeur d'une propriété et peut la définir sauf si « [r/o] » (lecture seule) apparaît après la syntaxe de la propriété.

RUBRIQUES CONNEXES :

Définitions de commandes
Définitions d'objets
Commandes AppleScript
Utiliser des dictionnaires AppleScript

OUVRIR :

Guide du langage AppleScript : Référence sur les classes

Définitions de commandes

Les commandes sont des termes dans AppleScript qui servent à exécuter des opérations. Les commandes d'un

dictionnaire AppleScript sont affichées dans la colonne centrale et sont désignées par la lettre « C » bleue. La définition d'une commande comprend une brève description du rôle de la commande et de sa syntaxe. La syntaxe est l'ensemble des règles déterminant l'écriture de la commande dans un script. Elle peut se composer de plusieurs parties.

Les mots en gras sont les termes utilisés dans une instruction. Vous devez taper ces termes tels qu'ils apparaissent dans le dictionnaire.

Les termes en bleu correspondent aux informations que vous fournissez. Ils peuvent servir par exemple à identifier un objet ou une variable que vous avez définie.

Des éléments facultatifs peuvent apparaître entre parenthèses dans les commandes. Il n'est pas nécessaire de les utiliser dans les instructions.

RUBRIQUES CONNEXES :

Définitions de classes
Utiliser des dictionnaires AppleScript
Définitions d'événements
Définitions d'objets

OUVRIR :

Guide du langage AppleScript : Référence sur les commandes

Définitions d'événements

Une définition d'événement décrit un verbe, ou commande, utilisé pour appliquer une action à un objet pilotable. Les événements sont les actions, ou verbes, que les objets peuvent accomplir ou auxquels ils peuvent participer.

Une définition d'événement commence généralement par le terme suivi d'une brève description du terme sur la même ligne.

Vient ensuite une liste de paramètres utilisés dans l'exécution de la commande. Les paramètres facultatifs sont indiqués entre crochets alors que les paramètres obligatoires ne le sont pas.

Chaque entrée de paramètre commence par le nom du paramètre suivi d'une description du format de données utilisé avec le paramètre et se termine par une brève description du paramètre.

La définition d'une commande comprend une brève description (correspondant au texte souligné) du rôle de la commande et de sa syntaxe. La syntaxe est l'ensemble des règles déterminant l'écriture de la commande dans un script. Elle peut se composer de plusieurs parties.

Les mots en gras sont les termes utilisés dans une instruction. Vous devez taper ces termes tels qu'ils apparaissent dans le dictionnaire.

Les termes au format Texte standard correspondent aux informations que vous fournissez. Ils peuvent servir par exemple à identifier un objet ou une variable que vous avez définie.

Des éléments facultatifs peuvent apparaître entre parenthèses dans les commandes. Il n'est pas nécessaire de les utiliser dans les instructions.

Les commandes agissent sur des objets. La définition d'un objet présente la syntaxe servant à identifier l'objet. Elle comprend son nom et une brève description (correspondant au texte souligné). La classe d'un objet détermine certaines caractéristiques de l'objet, par exemple les commandes que vous pouvez utiliser avec celui-ci.

RUBRIQUES CONNEXES :

Événements de base de données
Définitions de classes
Définitions de commandes
Définitions d'objets
Fonctionnement d'AppleScript

Définitions d'objets

Les commandes agissent sur des objets. La définition d'un objet dans un dictionnaire AppleScript présente la syntaxe servant à identifier l'objet. Elle comprend son nom et une brève description (correspondant au texte souligné). La classe d'un objet détermine certaines caractéristiques de l'objet, par exemple les commandes que vous pouvez utiliser avec celui-ci.

Si un objet a une forme plurielle, il vous permet de faire référence à plus d'un objet.

Si un objet fait possède une liste d'éléments, cela signifie qu'il contient d'autres objets. Le nom de l'élément, que vous pouvez utiliser dans votre script, apparaît en gras. Le texte standard décrit la méthode permettant de faire référence à l'élément.

Les objets peuvent avoir des propriétés. Le nom des propriétés apparaît en bleu et en gras. Le texte standard bleu décrit le type de valeur que contient la propriété, telle qu'une chaîne de caractères. Les propriétés peuvent être une valeur constante ou une liste de constantes parmi plusieurs possibles. Un script peut obtenir la valeur d'une propriété et peut la définir sauf si « [r/o] » (lecture seule) apparaît après la syntaxe de la propriété.

RUBRIQUES CONNEXES :

Utiliser AppleScript pour piloter des applications
Définitions de commandes
Définitions de classes
Attribuer une action à un objet
Utiliser des dictionnaires AppleScript

OUVRIR :

Guide du langage AppleScript : Référence sur les classes
Guide du langage AppleScript : Propriétés

Instructions If

Une instruction if est une commande uniquement exécutée lorsque certaines conditions sont réunies. Chaque instruction « if » contient une expression booléenne évaluée comme étant vraie ou fausse. Les commandes ainsi contenues dans l'instruction « if » sont exécutées seulement si la valeur de l'expression booléenne est Vrai. L'instruction « if » se termine par « end if ». Exemple :

```
if amountReceived > 100 then
        set memberStatus to "special"
end if
```

Dans l'instruction présentée ci-dessus, si l'expression booléenne est Vrai (la valeur du montant reçu est supérieure à 100), les commandes se trouvant dans l'instruction « if » sont alors exécutées. Si l'expression booléenne est Faux, les commandes ne sont pas exécutées.

L'instruction « if » peut également être construite avec l'expression réservée « else ». Cette association est connue sous le nom d'instruction if-then-else. Exemple :

```
if amountReceived > 100 then
        set memberStatus to "special"
else
        set memberStatus to "general"
end if
```

Dans l'instruction présentée ci-dessus, si la première expression booléenne est Vrai (la valeur du montant reçu est supérieure à 100), les commandes se trouvant dans la première partie de l'instruction « if » sont alors exécutées. Si la première expression booléenne est Faux, les commandes se trouvant dans la deuxième partie de l'instruction (c'est-à-dire après l'expression « else ») sont exécutées.

Vous pouvez imbriquer des instructions if afin de gérer plusieurs conditions.

Pour en savoir plus sur les instructions if dans AppleScript, consultez le guide « AppleScript Language Guide » (Guide du langage AppleScript).

RUBRIQUES CONNEXES :

Instructions AppleScript Repeat
Définitions de commandes
Définir des variables dans AppleScript
Valeurs constantes

OUVRIR :

AppleScript Language Guide - If

Instructions AppleScript Repeat

Vous pouvez utiliser des instructions repeat pour effectuer des tâches plusieurs fois en fonction de certaines conditions. L'instruction « repeat » permet d'exécuter en boucle un ensemble d'instructions jusqu'à ce qu'une condition soit remplie pour que l'exécution sorte de la boucle. Vous pouvez imbriquer des instructions repeat afin de réagir à plusieurs conditions.

AppleScript dispose d'une gamme d'instructions repeat pouvant répondre à différentes conditions.

repeat (forever)

Se répète en boucle à l'infini. Utilisez une instruction de sortie pour quitter la boucle.

```
repeat
        -- statement
        -- statement
        if (x = 5) then exit repeat
end repeat
```

repeat (number) times

Effectue le nombre de répétitions indiqué.

```
repeat 5 times
        -- statement
        -- statement
end repeat
```

---

repeat while

Effectue une répétition tant qu’une condition spécifiée est définie sur True.

```
set x to 1
repeat while (x < 5)
        -- statement
        -- statement
        set x to (x + 1)
end repeat
```

---

repeat until

Effectue une répétition jusqu’à ce qu’une condition spécifiée soit définie sur True.

```
set x to 1
repeat until (x = 5)
        -- statement
        -- statement
        set x to (x + 1)
end repeat
```

---

repeat with (variable) from (startValue) to (stopValue)

Définit la variable sur la valeur de départ, et augmente la valeur de la variable par incréments après chaque passage dans la boucle. La répétition se termine lorsque la variable atteint la valeur de sortie.

```
repeat with x from (5) to (25)
        -- statement
        -- statement
end repeat
```

---

repeat with (variable) in (list)

Définit la variable sur le premier élément de la liste. Après chaque passage dans la boucle, la variable est définie sur l’élément suivant. La répétition s’arrête lorsque tous les éléments de la liste ont été utilisés.

```
set myList to {234, 342, 681, 874}
repeat with x in myList
        -- statement
        -- statement
```

```
end repeat
```

Pour en savoir plus sur les instructions repeat AppleScript, consultez le « Guide du langage AppleScript » (en anglais).

RUBRIQUES CONNEXES :

Instructions If
Définir des variables dans AppleScript
Valeurs constantes

OUVRIR :

Guide du langage AppleScript : Repeat

Valeurs constantes

Une constante est un mot réservé dans AppleScript contenant une valeur déterminée à l'avance. AppleScript contient de nombreuses constantes utilisables pour obtenir des informations sur des objets, effectuer des comparaisons ou des calculs arithmétiques. Les applications et tout autre objet peuvent également définir leurs propres constantes. Ces constantes sont déterminées dans les définitions de classes de valeurs.

Pour consulter la valeur d'une constante, collez l'exemple suivant dans l'Éditeur AppleScript, puis cliquez sur Éxécuter. Cliquez ensuite sur Résultat pour afficher la valeur de la constante.

Le résultat est également une constante. Il constitue donc une constante issue d'un script faisant référence à la valeur renvoyée par une commande.

```
get the current date
```

Voici quelques autres exemples de constantes :

- constantes arithmétiques : pi ;
- constantes booléennes : true, false ;
- constantes de date et d'heure : minutes, heures, jours, semaines ;
- constantes de chaînes de caractères : retour, espace, tab ;
- constantes de scripts : anything, current application, it, me, missing value, my, result.

Pour en savoir plus sur les constantes AppleScript, consultez le Guide du langage AppleScript (en anglais).

RUBRIQUES CONNEXES :

Définir des variables dans AppleScript
Définitions d'objets
Définitions de classes
Définitions d'événements

OUVRIR :

Guide du langage AppleScript : Constantes

Définir des variables dans AppleScript

Une variable est un conteneur créé au sein d'un script destiné à stocker une valeur. Vous devez préciser un nom pour la variable, appelé également identifiant, au moment de sa création. Lorsque vous exécutez un script, vous avez la possibilité d'assigner une valeur à la variable, d'y accéder ou encore de la modifier. Les variables ne subsistent que pendant l'exécution du script. Lorsque ce dernier prend fin, les valeurs de ces variables sont perdues.

- Les exemples suivants montrent comment spécifier un identifiant unique et lui affecter une valeur.

```
set myVar to "153"

set x to "Quarterly Reports"

set x to false
```

Pour voir comment les valeurs sont stockées dans une variable, collez ces exemples dans l'Éditeur AppleScript, puis cliquez sur Éxécuter. Cliquez ensuite sur Résultat pour afficher la valeur stockée dans la variable.

Pour en savoir plus sur les variables AppleScript, consultez le Guide du langage AppleScript (en anglais).

RUBRIQUES CONNEXES :

Valeurs constantes
Instructions If
Instructions AppleScript Repeat

OUVRIR :

Guide du langage AppleScript : Variables

Définir l'emplacement d'une application

Sous Mac OS X, les applications sont regroupées sous forme de « paquets ». Un paquet est un répertoire spécial qui contient les dossiers et fichiers qui constituent l'application. Pour cette raison, il se peut que les commandes « path to » et « info for » renvoient des résultats différents pour une application.

Si vous utilisez la commande « path to » pour une application, le chemin se termine généralement par « .app ». Il est possible que les extensions de fichier ne s'affichent pas dans le Finder dans le nom du fichier, mais vous pouvez quand même utiliser des extensions pour identifier un fichier dans un script.

Si vous utilisez la commande « info for » pour obtenir des informations sur une application, la propriété Folder du résultat est généralement « true ». Vous pouvez obtenir plus d'informations sur le fichier et sur son nom en utilisant des propriétés supplémentaires du résultat, comme l'extension cachée, l'extension du nom ou le nom affiché.

RUBRIQUES CONNEXES :

path to
info for

Faire référence à des dossiers, des fichiers et des disques
choose application
Commandes de script du Finder

Commandes Finder

Faire référence à des dossiers, des fichiers et des disques

AppleScript vous permet de faire référence à des dossiers, des fichiers et des disques en indiquant uniquement leur nom, sans le chemin. AppleScript peut identifier l'emplacement d'éléments comme le dossier Documents, le bureau, le dossier de départ et le disque de démarrage.

- Cet exemple indique comment faire référence à un dossier spécifique par son nom dans un script.

```
set x to the path to the desktop folder
```

RUBRIQUES CONNEXES :

Faire référence au Finder dans des scripts
path to
info for
Commandes de script du Finder

Commandes Finder

Faire référence au Finder dans des scripts

Pour utiliser les commandes s'appliquant au Finder, utilisez une instruction tell comme vous le feriez pour une application.

- Voici un exemple utilisant l'instruction « tell » dans un script. L'instruction « tell » redirige la fonction « number of windows » vers « Finder app ».

```
tell application "Finder.app"
        return the number of windows
end tell
```

RUBRIQUES CONNEXES :

Finder
Attribuer une action à un objet
Faire référence à des dossiers, des fichiers et des disques
Définitions d'objets
Commandes de script du Finder

OUVRIR :

Guide du langage AppleScript - Object specifiers

Commandes de script du Finder

Voici quelques commandes pouvant servir avec le Finder :

activate

alias file

alias list

application file

clean up

close

column

Computer-object

container

copy

count

data size

delete

desktop window

desktop

disk

document file

duplicate

eject

empty trash

exists

file

file information

find

Finder

Finder window

folder

get

icon view options

information window

Internet location file

item

list view options

make

move

moving folder window for

open

package

path to

preferences

preferences window

print

quit

Faire référence à des dossiers, des fichiers et des disques

Faire référence au Finder dans des scripts

reveal

select

set

sort

trash

update

window

Commandes Finder

Commandes complémentaires de script standard

Les compléments de script ajoutent de nombreuses commandes utiles pouvant être incluses dans vos scripts.

adding folder items to

beep

choose application

choose color

choose file

choose file name

choose folder

choose from list

choose URL

clipboard info

close access

closing folder window for

current date

delay

dialog reply

display

display dialog

do shell script

download

FTP item

get eof

info for

list disks

list folder

load script

mount volume

offset

open for access

open location

opening folder

path to

POSIX path

random number

read

round

run script

say

scripting components

set eof

set the clipboard to

set volume

store script

summarize

system attribute

time to GMT

upload

URL

write